IN THE SPECIFICATION

Please amend the specification as follows:

Please cancel the current Abstract of the Disclosure, and replace it with the new Abstract of the Disclosure as shown in the enclosed replacement sheet.

Please amend the paragraph beginning at page 6, line 16 as follows:

The ~~commodotization~~ commoditization of networking has enabled cost effective distribution of computing, while at the same time increasing the need to protect oneself from malicious computing (e.g., viruses and attacks). Electronic privacy protection has emerged as a new requirement. If data could be encrypted and whatever computing that needs to be applied to the data could be transformed and split into two, a first portion applicable directly to the encrypted data giving encrypted results, and a second portion applicable to the encrypted results to give the same answer as applying the original logic on unencrypted data, then many of the privacy requirements could be addressed. However, it is not known how to apply general logic to encrypted data, in this fashion. We know how to apply interesting subsets of logic, and the subset of SQL logic where this model is applicable is our focus. Prior work has given techniques to be used for this purpose, but the problem of how to put these techniques together in an optimum manner has not been addressed. This application models and solves that optimization problem by 1) distinguishing data and operator level partitioning functions, 2) giving new query transformation rules, introducing a "round trip" server-to-client-to-server operator, and 3) a novel query plan enumeration algorithm. By means of an example, it is shown that significant performance improvements are possible from application of the techniques in this application.

Please amend the paragraph beginning at page 7, line 15 as follows:

In this illustration, there are three fundamental entities. A client computer 100 encrypts data and stores the encrypted data at a server computer 102 in an encrypted client database 104 managed by an application service provider 106. The encrypted client database 104 is augmented with additional information (which we call the index) that allows certain amount of query processing to occur at the server computer 102 without jeopardizing data privacy. The client computer 100 also maintains metadata 108 which is used by a query translator 110 for translating

the user query 112 into different portions, i.e., a query over encrypted data 114, for execution on the server computer 102, and a query over decrypted data 116, for execution on the client computer 100. The server computer 102 generates an encrypted intermediate results set 118a, which is transferred to the client computer 100 and stored as temporary results 120. The client computer 100 includes a query executor 122 that decrypts the temporary results 120 and performs the query over decrypted data 116, which may include a filtering or sorting operation, to generate an updated intermediate results set 118b, which is then re-encrypted and transferred back to the server computer 102. The server computer 102 completes its query processing on the re-encrypted intermediate results set 118b, in order to generate a new intermediate results set 118c, which is provided to the client computer 100 and stored as temporary results 120. Finally, the query executor 122 in the client computer 100 decrypts the temporary results 120 and performs the query over decrypted data 116 in order to generate actual results 124 for display 126 to the user.

Please amend the paragraph beginning at page 13, line 3 as follows:

Therefore, the partitioning and query processing strategy used in this application generalizes the approach proposed in [14] along two important directions. First, Q^S executes over the encrypted representation directly generating a (possibly) super-set [[)]] of results. Second, the results of Q^S are decrypted and further processed by the client computer 100 using Q^C to generate the answer to Q. We refer to the above partitioning of Q into Q^S and Q^C as operator level partitioning.

Please amend the paragraph beginning at page 39, line 10 as follows:

We have presented a new concept, data level partitioning, that delivers significant performance improvements for certain classes of queries. We have also introduced and formally studied a new communication scheme between the client computer 100 and server computer 102, which allows more [[that]] than one interaction between the client computer 100 and server computer 102 during query processing, whereas the previous work assumes that there is only a one time interaction. This new concept also allowed us to improve query execution plans substantially. We have conducted experimental tests to show the effectiveness of the schemes we presented in the application.